King Fahd University of Petroleum & Minerals Computer Engineering Dept

CSE 642 – Computer Systems Performance

Term 041

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

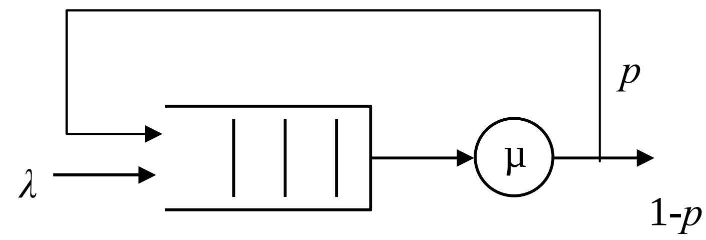
Email: ashraf@ccse.kfupm.edu.sa

Example: M/M/1 queue with Feedback

- **Problem**: Consider the following system – Find the pmf for number of customers in the system.

- **Solution**:

  $\Lambda = \lambda + p\,\Lambda \rightarrow \Lambda = \lambda/(1-p)$
  Therefore, traffic load, R is given by
  $R = \Lambda/\mu = \lambda/[\mu(1-p)]$
  $Prob(N = k) = (1-R)R^k \quad k=0, 1, 2, \ldots$



  Note $R < 1 \rightarrow \lambda/[\mu(1-p)] < 1$ or $\lambda < \mu(1-p)$ – this imposes a limit on the maximum arrival rate

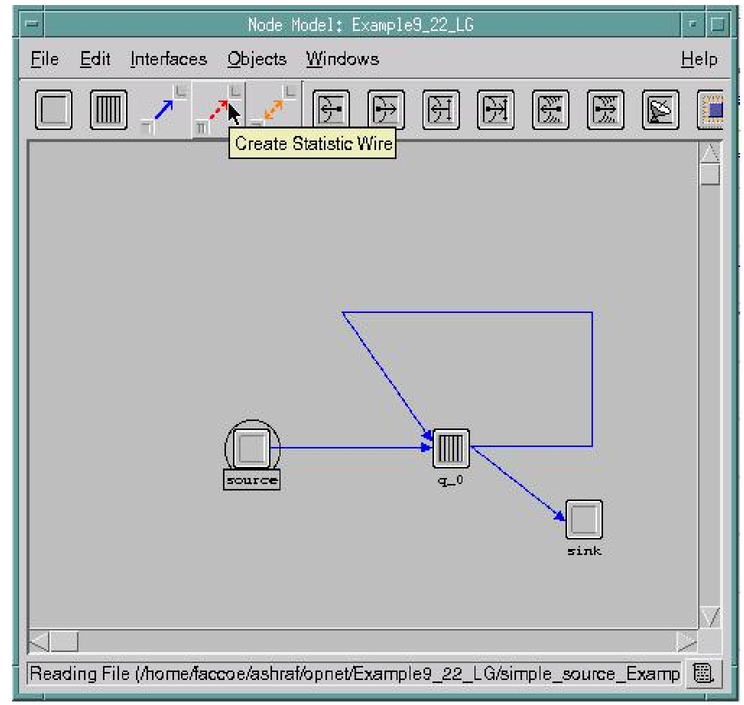
What is the average number of customer visits to the queue?

  $E[N] = R/(1-R)$
  $E[T] = E[N] / \lambda$ - direct application of Little's formula

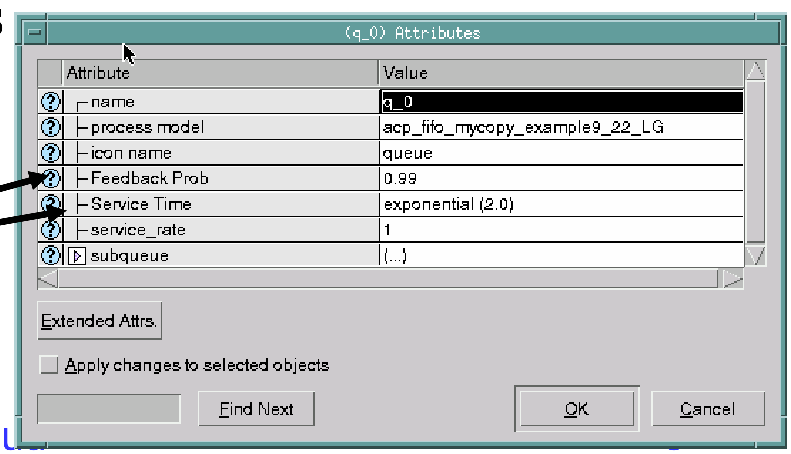
  For a general solution of an M/G/1 with Bernoulli feedback check: L. Takács, "A Single-Server Queue with Feedback," Bell Technical Journal, March 1963, pp. 505-519.

Example: M/M/1 queue with Feedback – Using Opnet

- **On the node model editor, build the following node**
  - **Use standard simple source, acp_fifo, sink process models for the shown objects.**
- **Standard acp_fifo need to be modified to allow for exponential service and also for providing a feedback route**
  - **Adding "Feedback Prob" attribute**
  - **Allowing for exponential service times – The original code assumes deterministic "constant" service time only**



- **These are new attributes**



Example: M/M/1 queue with Feedback – Modifying the Queue Process Model

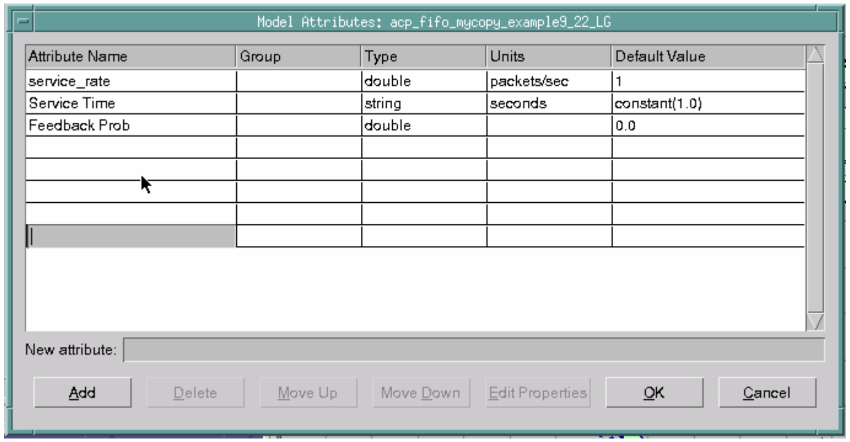
- **The figure shows the standard process model for acp_fifo**
- **To add the "Feedback Prob" attributed we do the following:**
  - **Interfaces/Model Attributes we add the "Feedback Prob" attribute of type "double" and default value of "0"**
  - **Open the temporary variables "TV" and add the following line:**

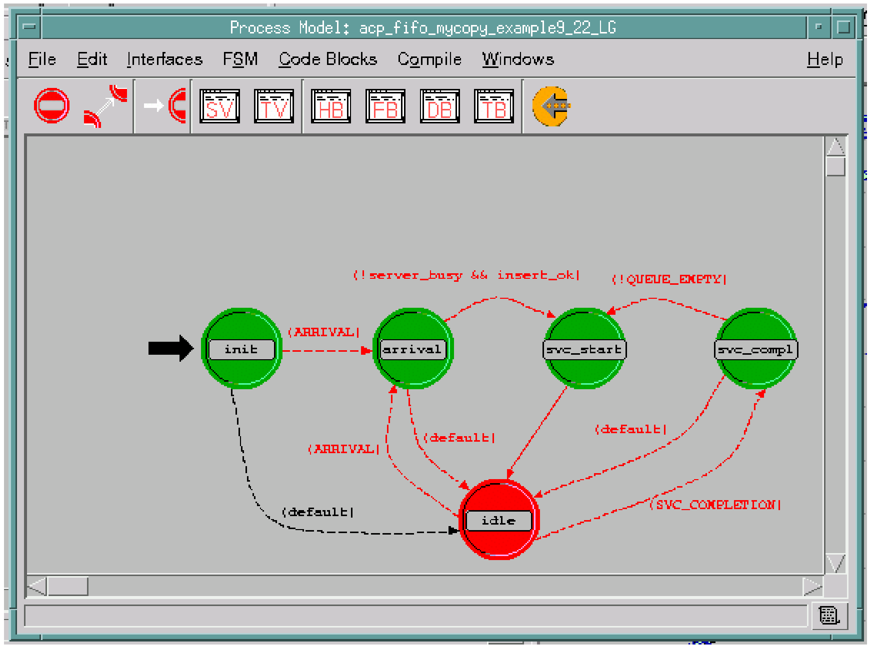
```
double p_random, FeedbackProb;
```

  - **Click on the entry part of the svc_compl state and replace the original** `pk_op_send_forced(pkptr, 0);` **statement with the following:**

```
p_random = op_dist_uniform(1.0);
    if (p_random <= FeedbackProb){
        pk_op_send(pkptr, 1);
    }
    else{
        pk_op_send_forced(pkptr, 0);
    }
```

  - **The above assumes is that port 1 is connected to the feedback stream while port 0 is the one connected to the sink (use "show connectivity" to know what port connected to which nodes).**



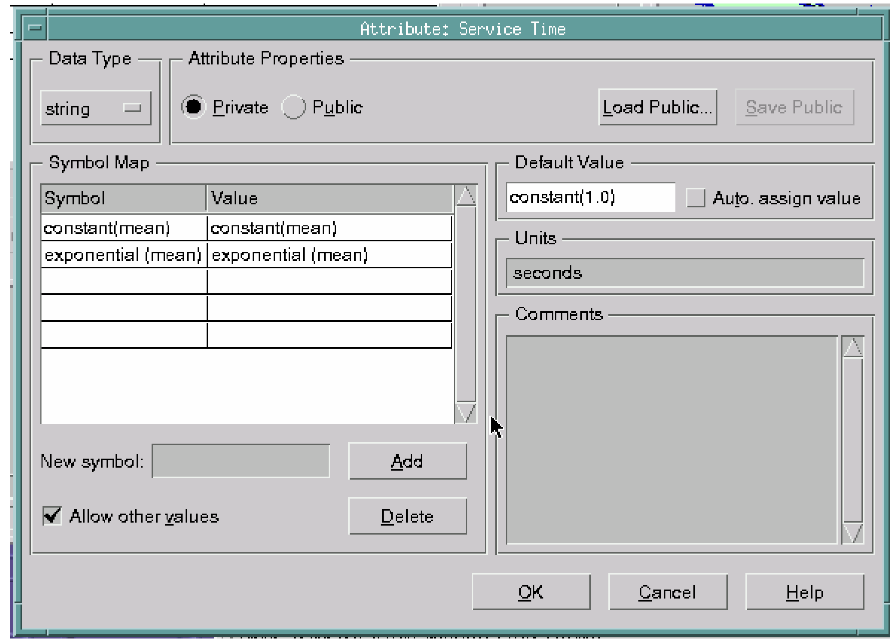


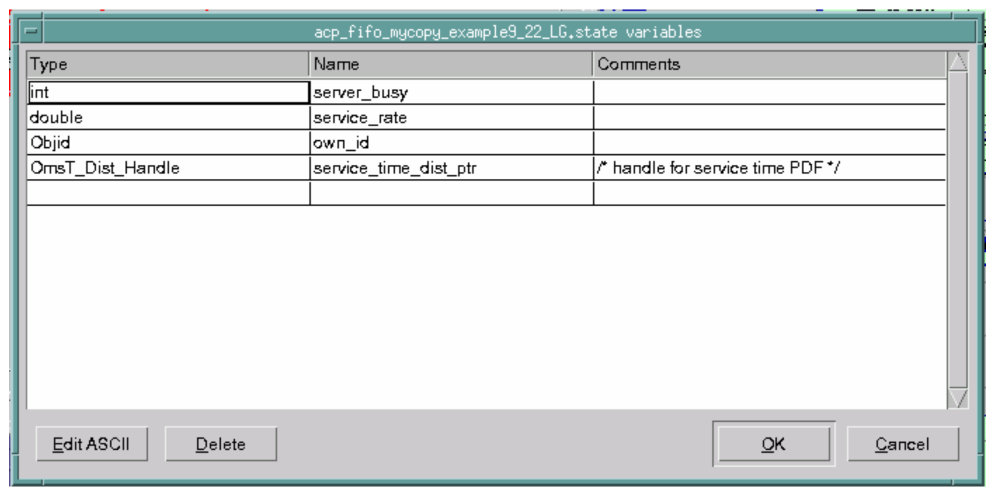
Example: M/M/1 queue with Feedback – Modifying the Queue Process Model

- **To allow for exponential service we define a new attribute called "Service Time" and we allow the attribute to be of constant or exponential time.**
- **We add the attribute "Service Time" through Interfaces/Model Attributes/Add (in the same manner we added the "FeedbackProb" attribute). The "Service Time" attribute should be of type string. Click on "Properties" and fill in the fields as shown in window.**
- **Click on the state variables "SV" button to add a state variable that will be used to store the distribution for the service time – the new state variable name is "`service_time_dist_ptr`" and is of type "`OmsT_Dist_Handle`"**
- **Click on the temporary variables "TV" button and add the following temporary variable definition "`char service_time_str[128]`"**
- **In the entry part of the init state you need to add code to read the attributes we have defined. The following code does exactly that:**

```
op_ima_obj_attr_get(own_id, "Service Time", service_time_str);
op_ima_obj_attr_get(own_id, "Feedback Prob", &FeedbackProb);
service_time_dist_ptr = oms_dist_load_from_string(service_time_str);
```

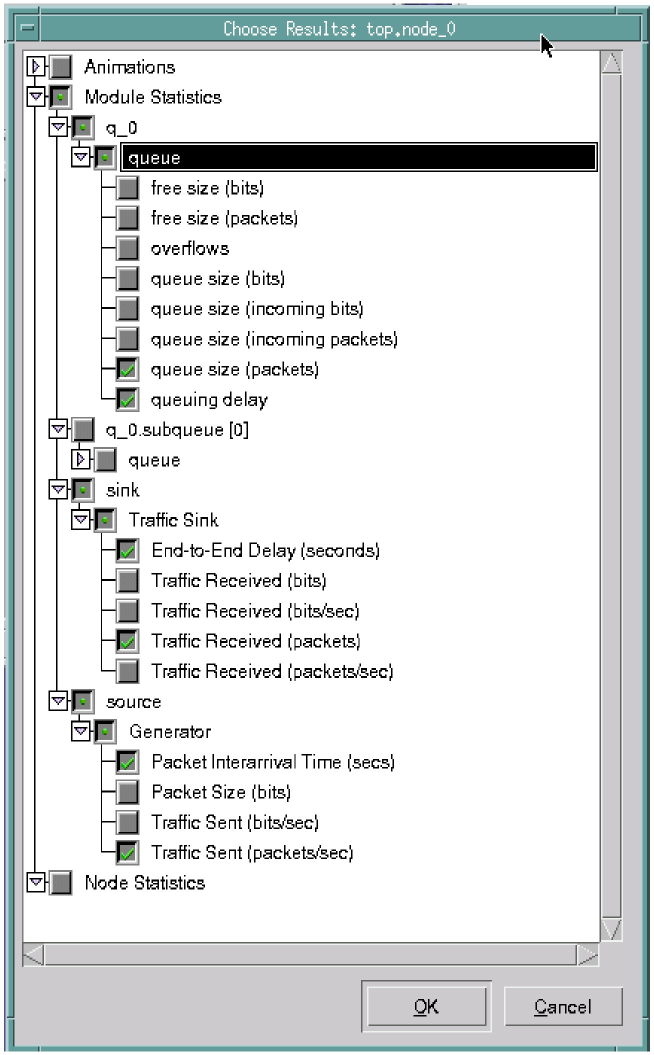
- **Finally, in the entry part of the svc_start state you need to comment the line "`pk_svc_time = 1.0`" and add the following line: "`pk_svc_time = oms_dist_outcome(service_time_dist_ptr);`"**





Example: M/M/1 queue with Feedback – cont'd

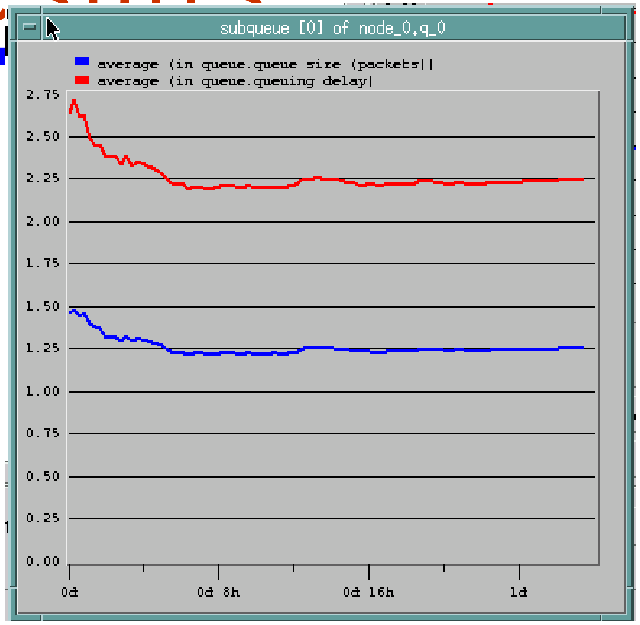
- **Before we run the simulation – save all your works spaces. Create a project scenario that includes the node model you created.**
- **To select which results to include in your simulation – right click on the project node, and choose "Choose Individual DES Statistics"**
- **Goto DES/Configure/Run DES for running simulation – in there you can specify the simulation time plus many other configuration parameters**
- **After the simulation is complete, you can view results by right clicking on the project icon and choosing "View Results"**
- **There are other ways to specify which statistic to collect and view – see the usage of scalar files next**

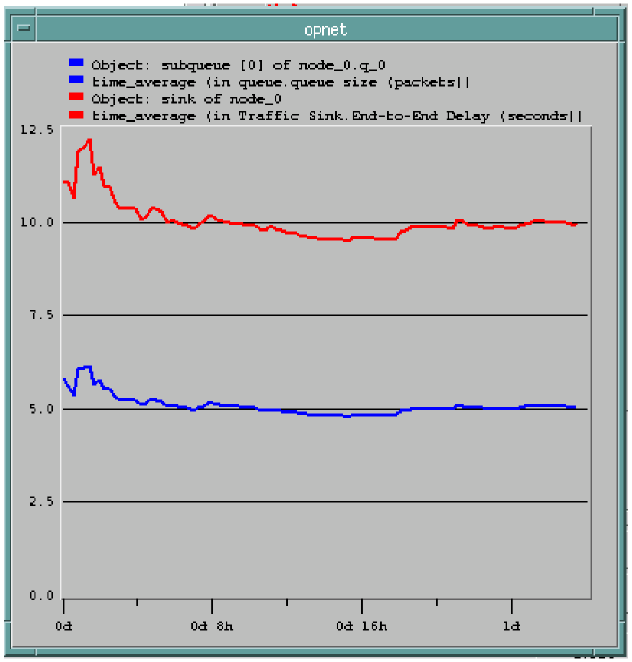


Example: M/M/1 queue with Feedback – Simulation Results

- **The simulation is run for the following parameters:**
  - **λ = 0.5 packets/second**
  - **μ = 1 packet / second**
  - **p =0.1 and 0.4**
- **The shown curves indicate that at steady state for p = 0.1**
  - **E[N] = 1.25**
  - **E[T] = 2.25 seconds**
- **The shown curves indicate that at steady state for p = 0.4**
  - **E[N] = 5.0**
  - **E[T] = 10.0 seconds**

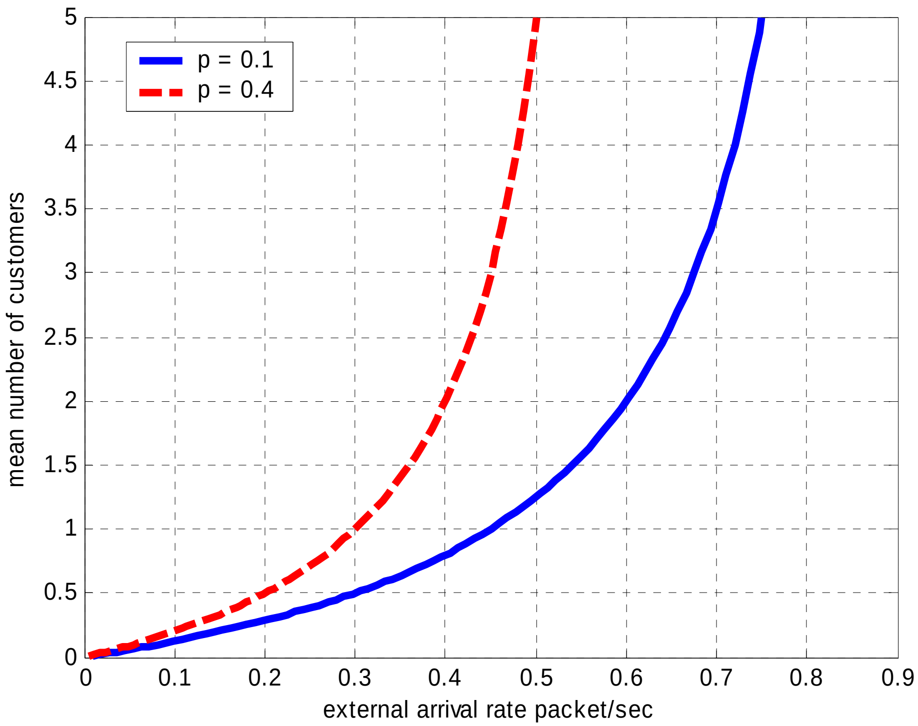
- **This form of results (statistic versus time) is know as a vector output – however usually for us, the whole curve is abstracted in one number.**

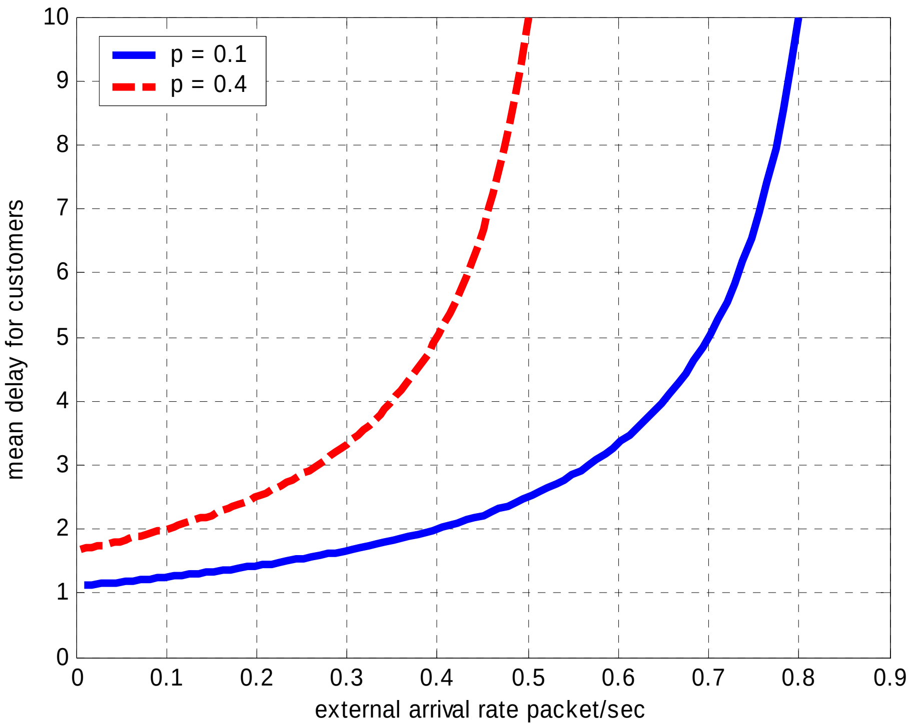




Example: M/M/1 queue with Feedback – Theoretical Results

- **The following graphs summarizes theoretical results --**





Example: M/M/1 queue with Feedback – Scalar Outputs and Simulation Sequences

- **Now we are in a position to run simulation for a series of loads to produce curves similar to the one produced theoretically. To do that we follow these steps:**
  - **Define a probe model**
  - **Define a sequence of simulations**
  - **Results will be written to a scalar file**

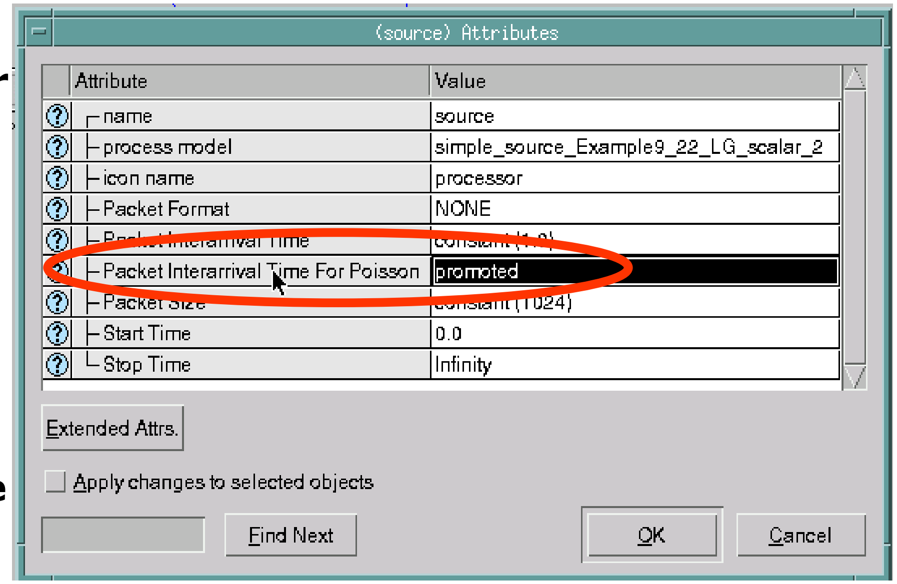
Example: M/M/1 queue with Feedback – Promotion of Attributes

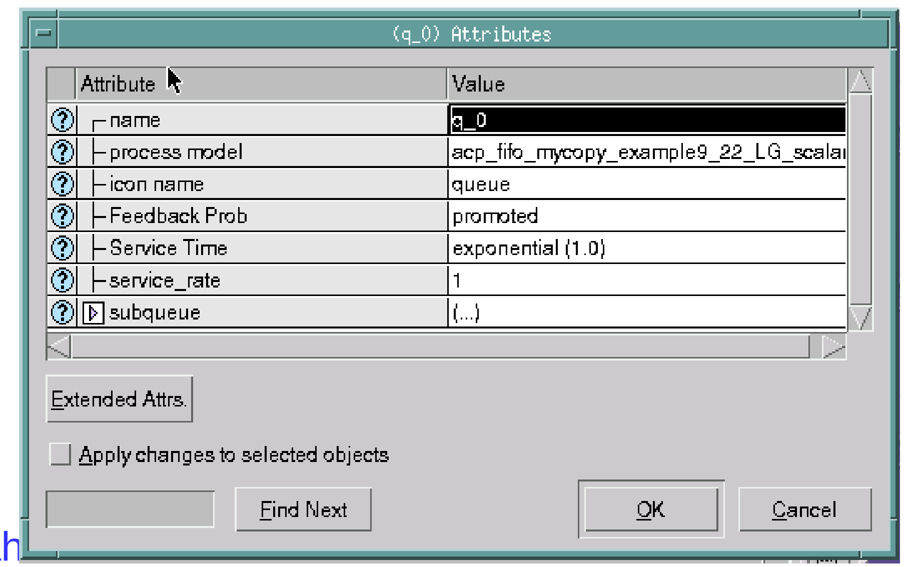
- **Since our we need to run multiple simulations for different interarrival times for incoming stream and different feedback probablities, we will need to "promote" these variables so that they can be accesses and initialized from the DES Configure/Run window.**

- **To do that open the node mode**
  - **Add another attribute for the source node called "Packet Interarrival Time for Poisson" and modify the code so that this is the variable used to determine the next arrival.**
  - **Right click on the source and open the attributed window. Right click on the value for the "Packet Interarrival Time For Poisson" and change it to promoted as shown in the figure.**
  - ***Why are we creating interarrival variable?***

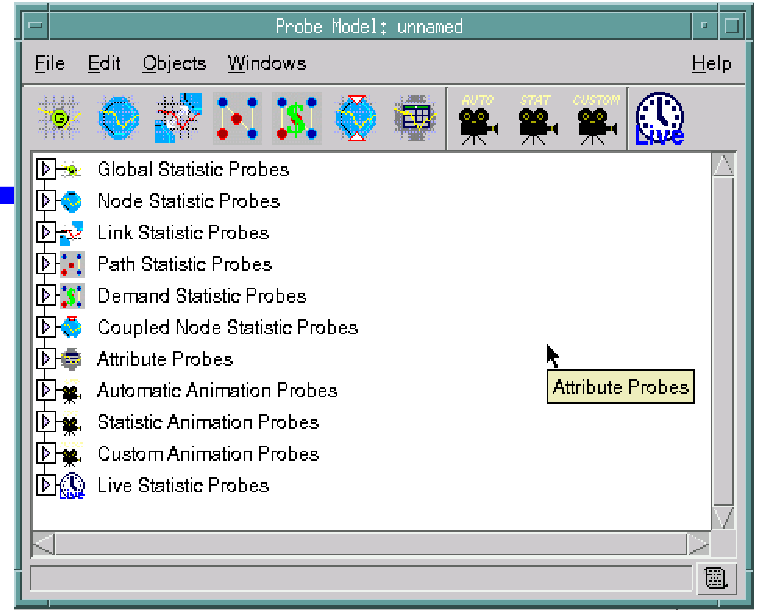
- **To promote the "Feedback Prob" attribute we do the same using the attributes for the q_0 node. The window is also shown.**

- **This will allow us to set multiple values for these variables**

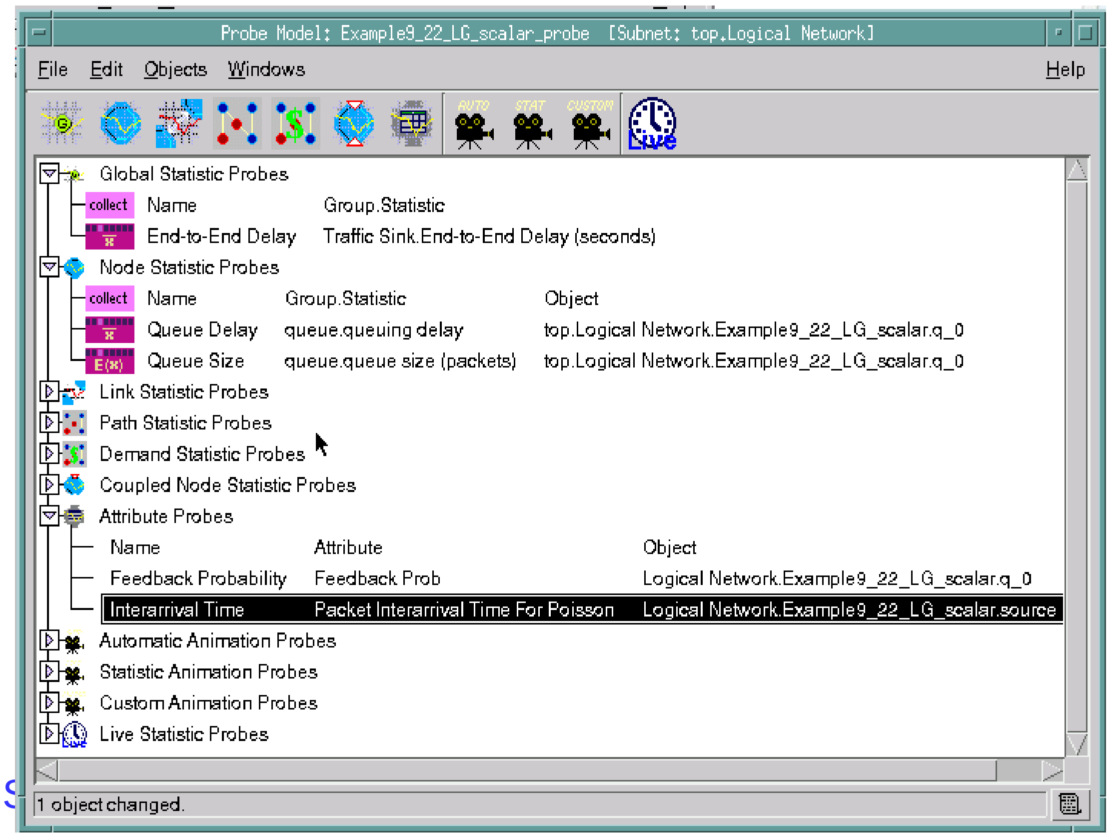




Example: M/M/1 queue with Feedback – Creating a Probe Model

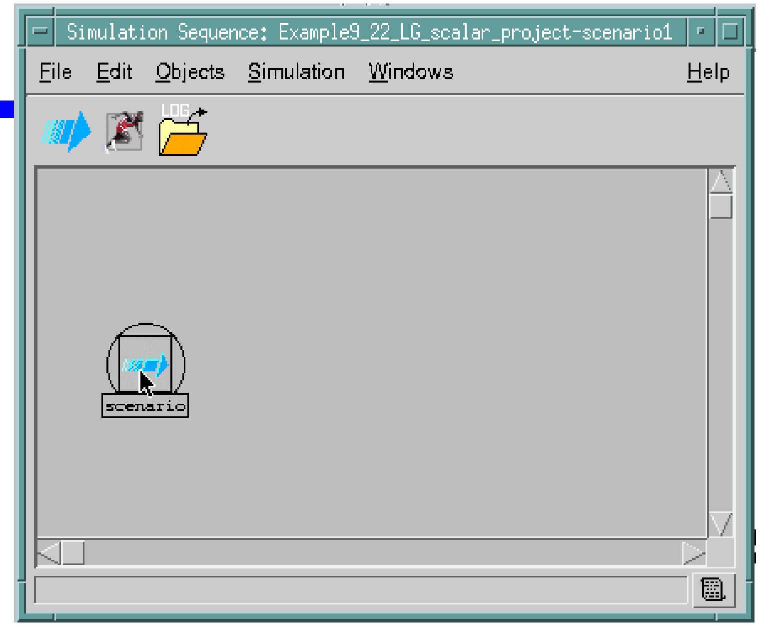


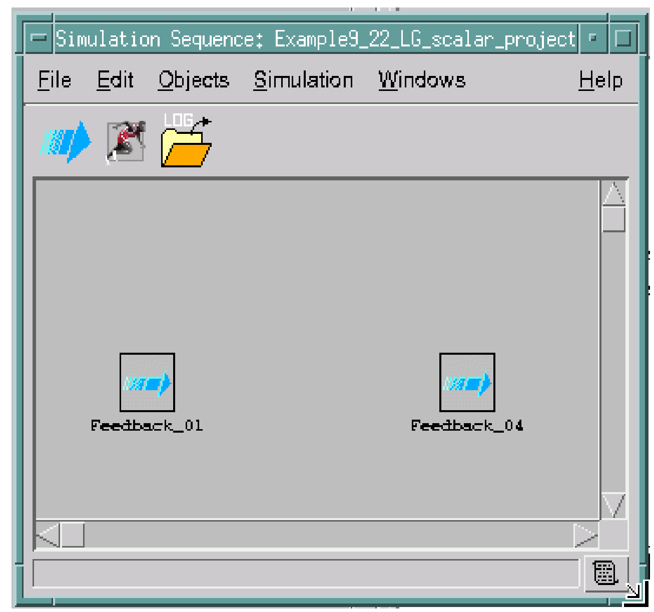
- **Probe Model is the model responsible for collecting the results from every simulation run.**
- **To create a probe model for our simulation, from the project window goto File/New/Probe Model – you get an unamed probe model similar to the one shown on the side**
- **Goto to Objects/Set network model and select our project in order to probe some of its results**
- **Now you are in a position to select probes for our statistics**
    - **Add a global statistic probe – and set it to probe the end-to-end delay**
    - **Add node statistic probes – and set them to probe the queue delay and the queue size**
    - **Add Attribute probes – and set them to probe the feedback probability and the interarrival time**
- **The final probe window should look like the one depicted on the side**



Example: M/M/1 queue with Feedback – Creating a Simulation Sequence

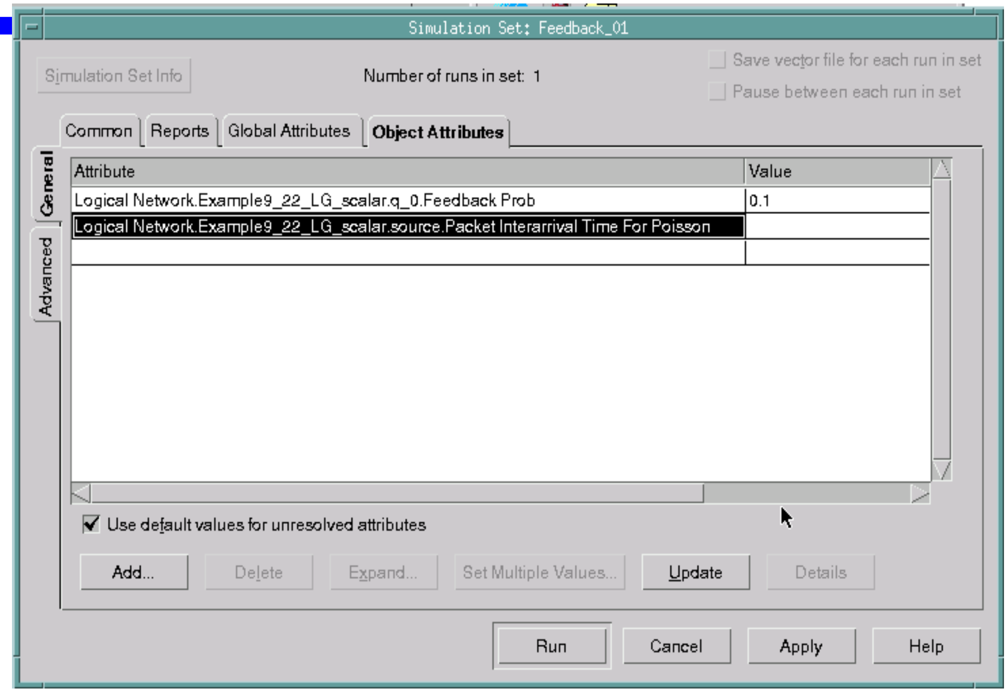
- **On the project window – select DES → Configure/Run DES and a window similar to the one on the side appears.**
- **Use Edit/copy/paste to create another icon. Right click on these icons and name them "Feedback_01" and "Feedback_04". This is achieved by right clicking on the icon – editing the attributes and setting the simulation set name to the name "Feedback_01" or "Feedback_04".**
- **The "Feedback_01" icon will be used to run a sequence of simulations for the feedback Prob = 0.1 scenario, while the "Feedback_04" icon will be used to run a sequence of simulations for the feedback Prob = 0.4 scenario**
- **The resulting window should look like the one shown on the side.**

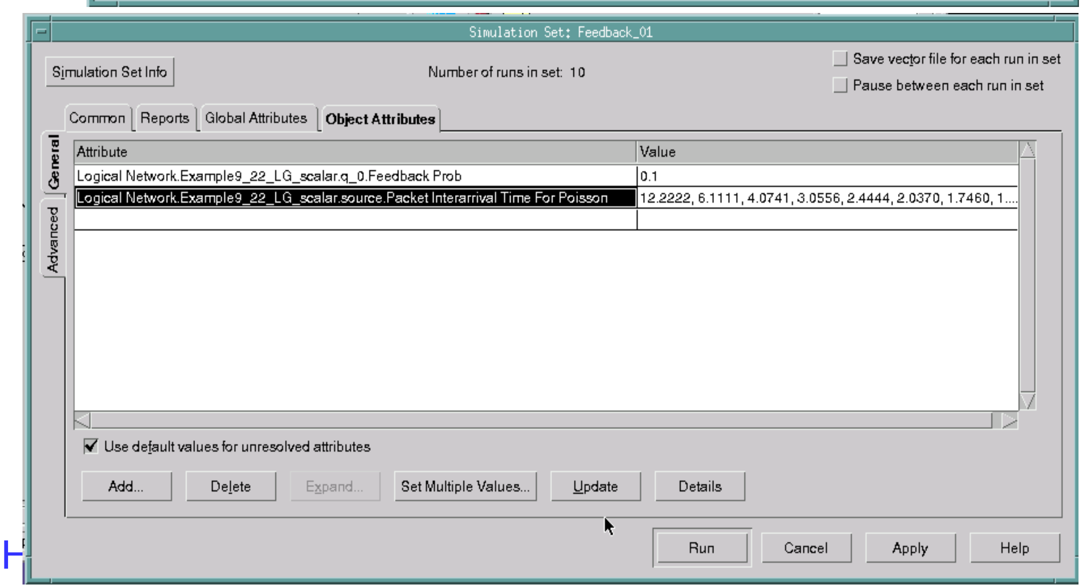




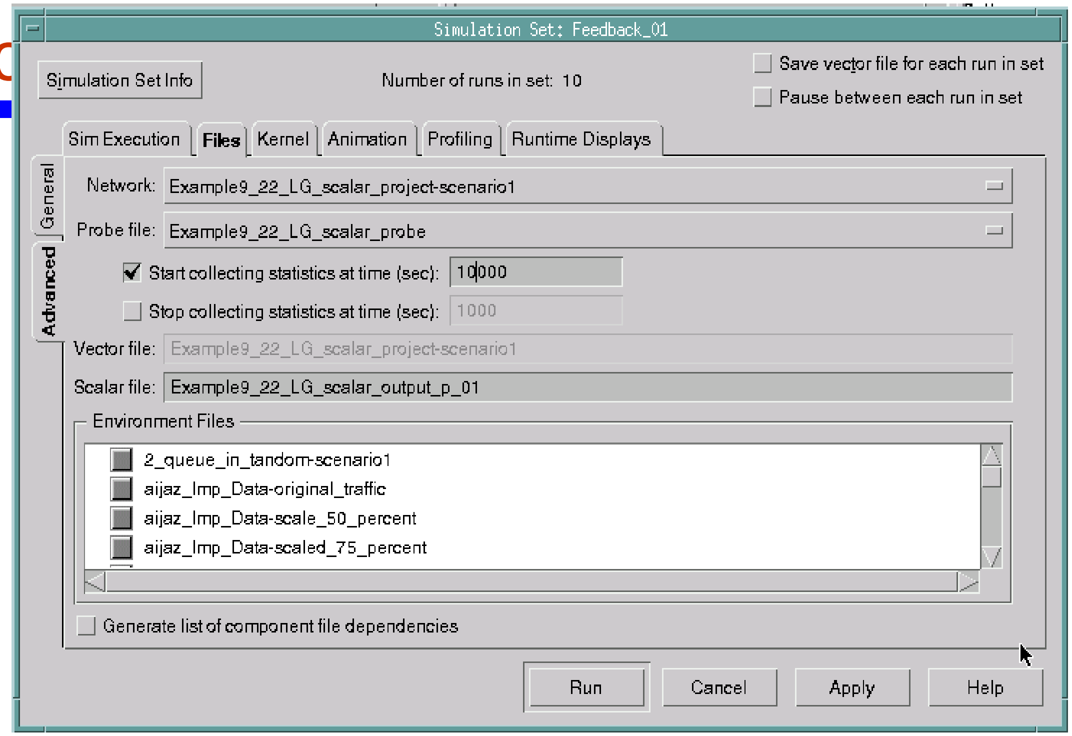
Example: M/M/1 queue with Feedback – Creating a Simulation Sequence – cont'd

- **To set the "Feedback_01" sequence – right click on the corresponding icon, go to the General/Common tab – set the simulation time to 500,000 seconds**
- **To initialize the simulation variables, go to the General/object attributes tab**
- **In this place you will have to provide values for the promoted attributes. Click on add and you will find out the two promoted nodes attributes we made in the previous slide. Select the "Feedback Prob" attribute. In the value column give it the initial value 0.1.**
- **Click add again and select the "Interarrival Time For Poisson" attribute and press OK. This create a window which looks like the one shown on the side**
- **For the interarrival time, we would like to give multiple values (as a reflection for the variation of the load)**
- **Remember that the rate of arrivals is upper bounded by $\mu(1-p)$ – therefore, $\lambda max\_01 <$ 0.9 and $<$ 0.6 for the cases $p = 0.1$ and 0.4, respectively.**
- **Say we desire 10 equally distance points on the arrival rate access, then for $p = 0.1$ we must use the following interarrival times: [12.2222, 6.1111, 4.0741, 3.0556, 2.4444, 2.0370, 1.7460, 1.5278, 1.3580, 1.2222] – to enter these use the "Set Multiple Value" button.**
- **Press Apply to store your work**





Example: M/M/1 queue with Feedback – Creating a Simulation Sequence – c



- **Now go to the Advanced/Files tab (shown in the figure)**
  - **Choose the probe model we created for this simulation**
  - **Tick the start time for collecting stats to 10000 seconds to eliminate transient behavior**
  - **Specify the scalar file name that will contain the results.**
  - **The overall setting is as shown in the figure on the side.**
- **Press Apply to store your work and then Cancel to exit the attribute editing for "Feedback_01" sequence.**
- **Now the same for the "Feedback_4" sequence except that**
  - **Feedback Prob = 0.4**
  - **Interarrival times = [18.3333, 9.1667, 6.1111, 4.5833, 3.6667, 3.0556, 2.6190, 2.2917, 2.0370, 1.8333]**
  - **Choose a different scalar file name for the outputs for this sequence**
- **Press Apply**
- **Now you sequence can be run by pressing Run – note you will be prompted whether you are sure you want to run 20 simulations or not**

Example: M/M/1 queue with Feedback – Converting Output Scalar File to Text Files

- **The output of the two runs is stored in to scalar files: Example9_22_LG_scalar_p_01.os and Example9_22_LG_scalar_p_04.os (note the .os extension)**
- **We can opnet's Analysis and Configuration tool to analyze and plot these results**
- **However, I will export the results to matlab!**
- **To convert scalar files to text use the following opnet command: op_cvos –to_text –m fname; where fname is the scalar file name without the ".os" extension.**
    - **If you do not get the output message "Converting .... Writing output file – Done" the conversion was not completed. "Reading input file" message alone does not mean anything!**
- **The previous command line converts the files to ".gdf" files which are text files and can be processes by excel or matlab**
- **The format of the gdf file (Example9_22_LG_scalar_p_01.gdf) is shown in the next slide**
    - **Note that results for each simulation run is called a data block**
    - **We have 10 data blocks (10 interarrival times)**
    - **Each data block contains the 5 probed statistics**
    - **The required value is separated by ":" from the label of the statistics – one can be used to facilitate loading the file into excel (i.e. use ":" as a column separator)**
    - **Data blocks are separated by 7 lines – this can be used when writing your own parsing code to extract the data.**

Example: M/M/1 queue with Feedback – Format of GDF/Output Scalar Files

```
total number of blocks: 10

BEGIN_BLOCK
model: Example9_22_LG_scalar_project-scenario1
seed: 128
--------------------
number of scalar records: 5
Feedback Probability:          0.1
Interarrival Time:12.2222
Queue Delay.mean:              1.09977127892
Logical Network.Example9_22_LG_scalar.q_0.queue.queue size (packets).average:     0.100610997419
Traffic Sink.End-to-End Delay (seconds).mean:     1.22349043036
```

1st point

```
BEGIN_BLOCK
model: Example9_22_LG_scalar_project-scenario1
seed: 128
--------------------
number of scalar records: 5
Feedback Probability:          0.1
Interarrival Time:6.1111
Queue Delay.mean:              1.21194500548
Logical Network.Example9_22_LG_scalar.q_0.queue.queue size (packets).average:     0.219311306834
Traffic Sink.End-to-End Delay (seconds).mean:     1.34544225305
```

2nd point

```
BEGIN_BLOCK
model: Example9_22_LG_scalar_project-scenario1
seed: 128
--------------------
number of scalar records: 5
Feedback Probability:          0.1
Interarrival Time:4.0741
Queue Delay.mean:              1.36937205236
Logical Network.Example9_22_LG_scalar.q_0.queue.queue size (packets).average:     0.372269219689
Traffic Sink.End-to-End Delay (seconds).mean:     1.52133542716
.
.
.
```

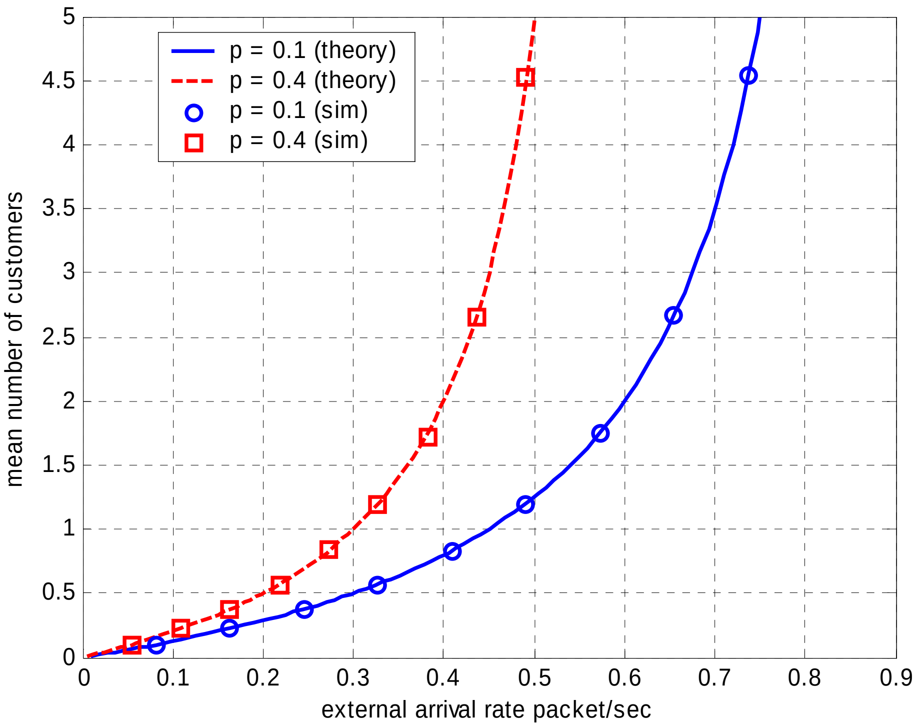
3rd point

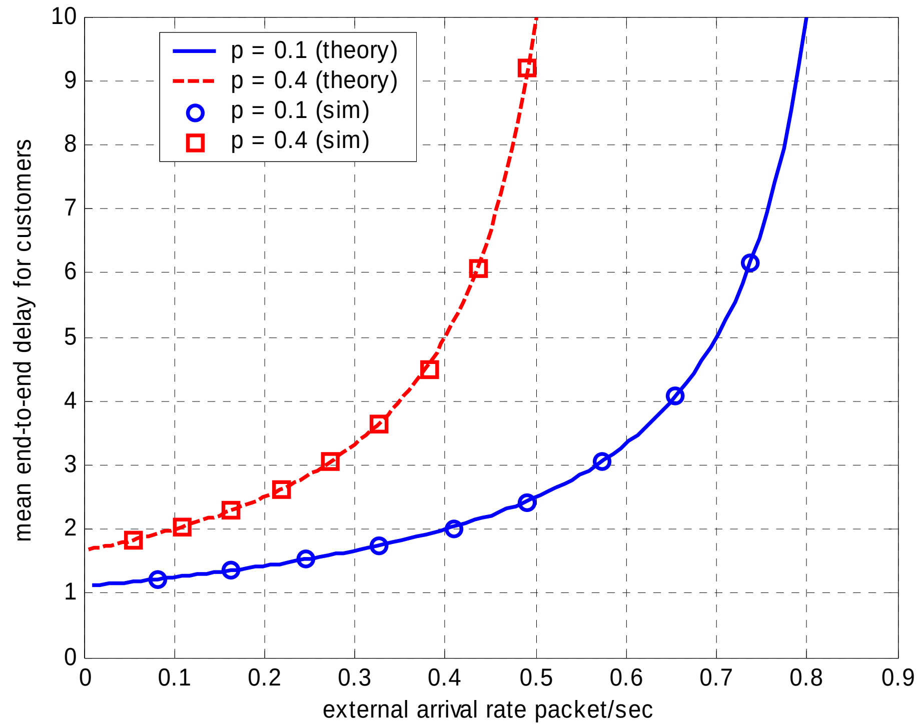
.

.

Example: M/M/1 queue with Feedback – Simulation Results (Finally!!)

- **The following two curves show the simulation results imposed on the previously shown theoretical results (refer to slide 30).**
- **Of course we can see an excellent match between theoretical results and simulation results**





Example: M/M/1 queue with Feedback – Simulation Results – Matlab Code (1)

- **The code used for processing the GDF files and plotting the previous results is as shown**

Code to Process GDF file:

```
function [P_feedback, InterarrivalTime, q_D, q_N, T] = ProcessGDF_Ex9_22_LG( fname );
%
% script to read opnet gdf files
fid = fopen(fname, 'r');
tline = fgetl(fid); % first line contain number of points
% read number of points
index      = find(tline == ':');
NoOfPoints = str2num(tline(index+1:length(tline))); clear tline
%
% discard following lines
for(i=1:1:6); tline = fgetl(fid); end;
for i=1:NoOfPoints
    %
    % read 5 variables
    tline = fgetl(fid); % feedback line
    index = find(tline == ':');
    P_feedback(i) = str2num(tline(index+1:length(tline))); clear tline
    tline = fgetl(fid); % interarrival-time line
    index = find(tline == ':');
    InterarrivalTime(i) = str2num(tline(index+1:length(tline))); clear tline
    tline = fgetl(fid); % queue delay line
    index = find(tline == ':');
    q_D(i) = str2num(tline(index+1:length(tline))); clear tline
    tline = fgetl(fid); % queue size line
    index = find(tline == ':');
    q_N(i) = str2num(tline(index+1:length(tline))); clear tline
    tline = fgetl(fid); % end_2_end delay line
    index = find(tline == ':');
    T(i)  = str2num(tline(index+1:length(tline))); clear tline
    %
    % discard following lines
    for(i=1:1:7); tline = fgetl(fid); end; clear tline
end
fclose(fid);
```

Example: M/M/1 queue with Feedback – Simulation Results – Matlab Code (2)

- **The code used for generate theoretical results**

Code to generate theoretical results:

```
function [Lambda, N_avg, T_avg] = One_Queue_WithFeedback( p, M, L_No)
%
% one queue with feedback with p
% take p - feedback probability
%      M - service rate (exponential)
%      N - number of points
% returns N points of:
%      Lambda - arrivals rates
%      N_avg - average queue size
%      T_avg - average end-to-end delay
L_max = M*(1-p);
L_Step = L_max/L_No;
Lambda = L_Step:L_Step:L_max-L_Step;
R      = Lambda./(M*(1-p));

N_avg = R./(1-R);
T_avg = N_avg./Lambda;
```

Example: M/M/1 queue with Feedback – Simulation Results – Matlab Code (3)

- **The code used for generating curves**

Code to plot results:

```
%
% one queue with feedback with p
% process and plot simulation results from files:
%   Example9_22_LG_scalar_output_p_01.gdf, and
%   Example9_22_LG_scalar_output_p_04.gdf
%
LineWidth = 2;
fname = 'Example9_22_LG_scalar_output_p_01.gdf';
[P_feedback_01, InterarrivalTime_01, q_D_01, q_N_01, T_01] = ProcessGDF_Ex9_22_LG( fname );
Lambda_01_sim = 1./InterarrivalTime_01;
fname = 'Example9_22_LG_scalar_output_p_04.gdf';
[P_feedback_04, InterarrivalTime_04, q_D_04, q_N_04, T_04] = ProcessGDF_Ex9_22_LG( fname );
Lambda_04_sim = 1./InterarrivalTime_04;
%
% Compute theoretical results
M = 1;
p = 0.1;
[Lambda_01, N_avg_01, T_avg_01] = One_Queue_WithFeedback( p, M, 100);
p = 0.4;
[Lambda_04, N_avg_04, T_avg_04] = One_Queue_WithFeedback( p, M, 100);
%
% plot queue size results
figure(1);
h = plot(Lambda_01, N_avg_01, '-',Lambda_04, N_avg_04, '--r', ...
         Lambda_01_sim, q_N_01, 'ob', Lambda_04_sim, q_N_04, 'sr');
set(h, 'LineWidth', LineWidth);
ylabel('mean number of customers');
xlabel('external arrival rate packet/sec');
axis([0 0.9 0 5]);
legend('p = 0.1 (theory)', 'p = 0.4 (theory)', 'p = 0.1 (sim)', 'p = 0.4 (sim)', 0);
grid;
%
% plot end-to-end delay results
figure(2);
h = plot(Lambda_01, T_avg_01, '-',Lambda_04, T_avg_04, '--r', ...
         Lambda_01_sim, T_01, 'ob', Lambda_04_sim, T_04, 'sr');
set(h, 'LineWidth', LineWidth);
ylabel('mean end-to-end delay for customers');
xlabel('external arrival rate packet/sec');
axis([0 0.9 0 10]);
legend('p = 0.1 (theory)', 'p = 0.4 (theory)', 'p = 0.1 (sim)', 'p = 0.4 (sim)', 0);
grid;
```

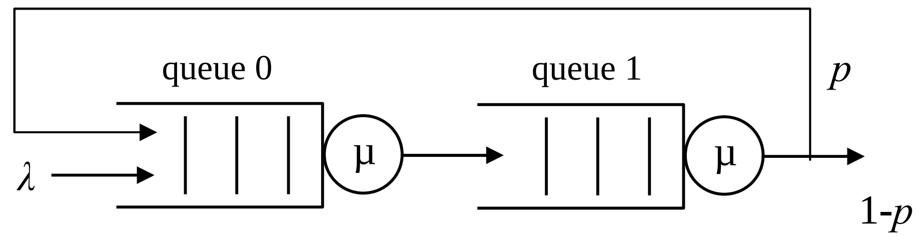
Assignment # 3

- **Use example 9.23 of Leon Garcia's textbook to do the following:**
  - **(a) Using the theoretical analysis supplied in the solved example in the textbook:**
    - **Plot the average total number of customers in network versus the external arrival rate**
    - **Plot the average end-to-end delay of a customer versus the external arrival rate**
  - **(b) Develop an opnet simulation model and produce simulation results and compare them against those obtained in part (a)**
    - **Produce comparative curves similar to those found on slide 39.**
  - **For part (a) and (b) use p = 0.9 and p = 0.6 (note p is the probability of exiting the network from queue 1)**

- **Deadline: Sunday January 2^{nd}, 2005 – class time**

Feedback Violates the Poisson Departure process

- **Let us examine the following system***



- **<u>Solution:</u> The analytic solution for the depicted system is as follows:**

$\Lambda = \lambda + p\,\Lambda \Rightarrow \Lambda = \lambda/(1-p)$

Therefore, traffic load, R is given by

$R_0 = R_1 = R = \Lambda/\mu = \lambda/[\mu(1-p)]$

$Prob(N_0 = k) = Prob(N_1 = k) = (1-R)R^k \quad k=0, 1, 2, \ldots$

Note $R < 1 \Rightarrow \lambda/[\mu(1-p)] < 1$ or $\lambda < \mu(1-p)$

$E[N_0] = E[N_1] = R/(1-R) = \lambda/[\mu(1-p) - \lambda]$

$E[N] = E[N_0] + E[N_1] = 2R/(1-R) = 2\lambda/[\mu(1-p) - \lambda]$

End-to-end delay for a customer is computed as

$E[T] = E[N] / \lambda = 2/[\mu(1-p) - \lambda]$

*I can show the same behavior using the single-server queue system considered in the last example – but I am using the system proposed the textbook to give another example on opnet modeling

Feedback Violates the Poisson Departure process – cont'd

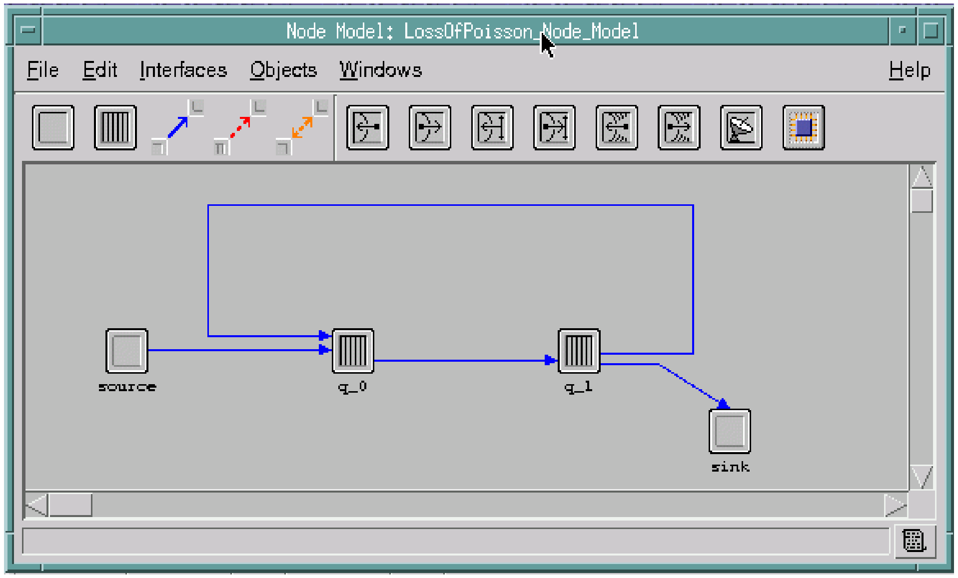
- **To depict the violation of the Poisson arrival/departure process**
  - **Assume a very low external arrival rate λ – say 1 packet every 2 hours – i.e. mean interarrival time of 7200 seconds**
  - **Assume a very small mean service time 1/μ – say 10^{-9} second**
  - **Let p = 0.999**
- **This setting translates the following:**
  - **One customer arrives – the next arrival is 1000s of seconds away on average**
  - **The customer is TRAPPED in the system circulating (since p ≈ 1**
  - **So if we monitor the customer departures of queue 0 or 1 (prior to the feedback branching) – we expect to see departure bursts**

Feedback Violates the Poisson Departure process – Using Opnet

- **Let us try to use opnet to check the aforementioned behavior described**
- **Create a new directory - open the node model designed for the single server queue and save it in new directory. This directory will contain all the required modules for this example**
- **Double click on each of the node to open the process model – save that too in the new directory**
- **Finally on node model, right click on the nodes and open the edit attributes – change the process model name to the one you saved in the directory for this example.**
- **The above steps guarantee that changes done to process models in this example will not affect other projects**
- **In the single-server queue model we built a general purpose queue node with the capability to provide feedback – we will use that to build the model for the current system**

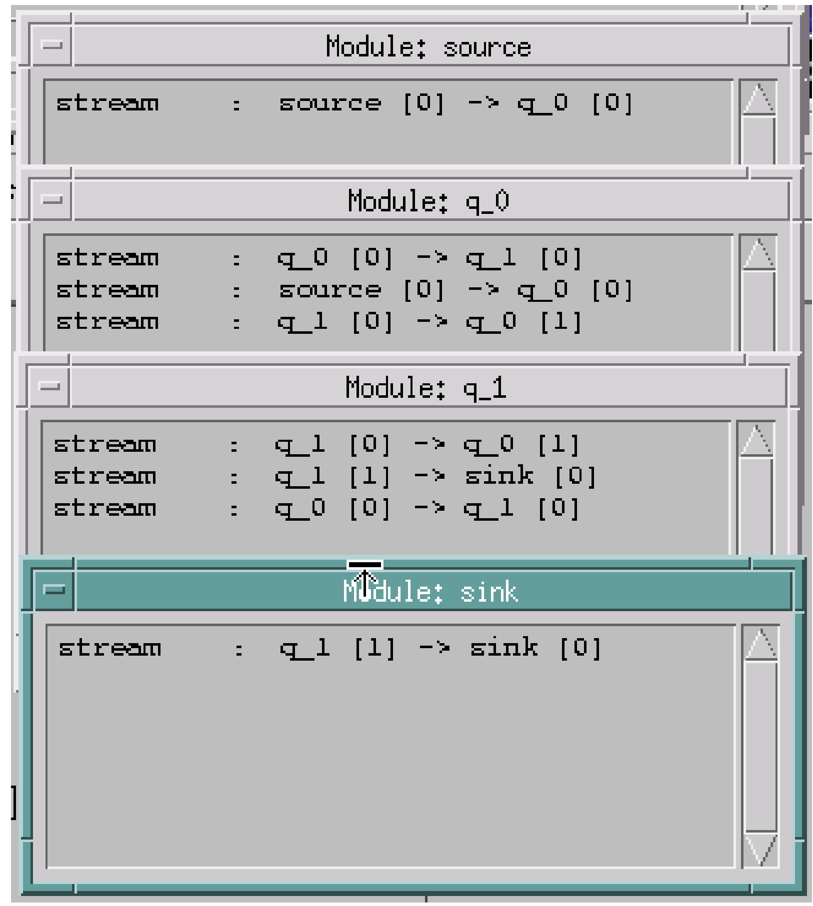
Feedback Violates the Poisson Departure process – Node Model

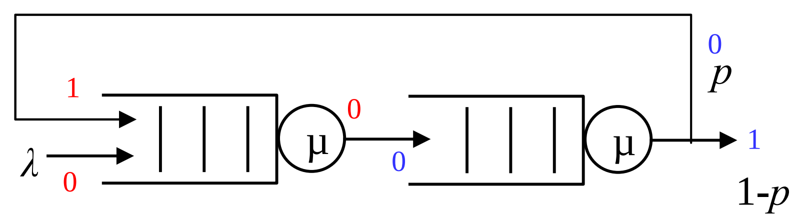
- **Modify the node model to look like the one depicted on the side figure**
  - **Click on the original queue node and then copy/paste to produce an identical node**
  - **Modify the connectivity to reflect the requirements for the current system**
  - **Note q_0 does not have a feedback branch!**



Feedback Violates the Poisson Departure process – Connectivity of Nodes

- **To show the connectivity of any node in your node model, right click on the node and select "Show Connectivity"**
- **The connectivity of our nodes is displayed in side figure.**
    - **The source node has one output port (#0) and is connected to the input port (#0) of node q_0**
    - **The q_0 node has two input ports (#0) and (#1) – the input port (#1) is coming from the output port #0 of node q_1**
    - **Note q_0 has only one output port (#0) connected to the input port #0 of node q_1**
    - **q_1 has only input port (#0) and two output ports (#0) connected to the input port (#1) of node q_0, and (#1) connected to the input port (#0) of the sink**
    - **The sink node has one input port #0 that is fed from the output port #1 of node q_1**
- **The input/output ports of nodes q_0 and q_1 are shown in the figure on the side**
- **The numbering of the ports may depend on the order of creating the packet streams – therefore you may have a different one that I do here !!**

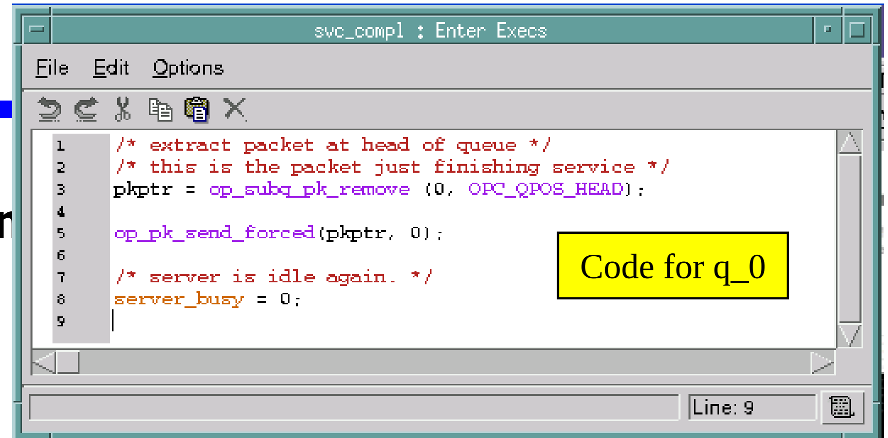


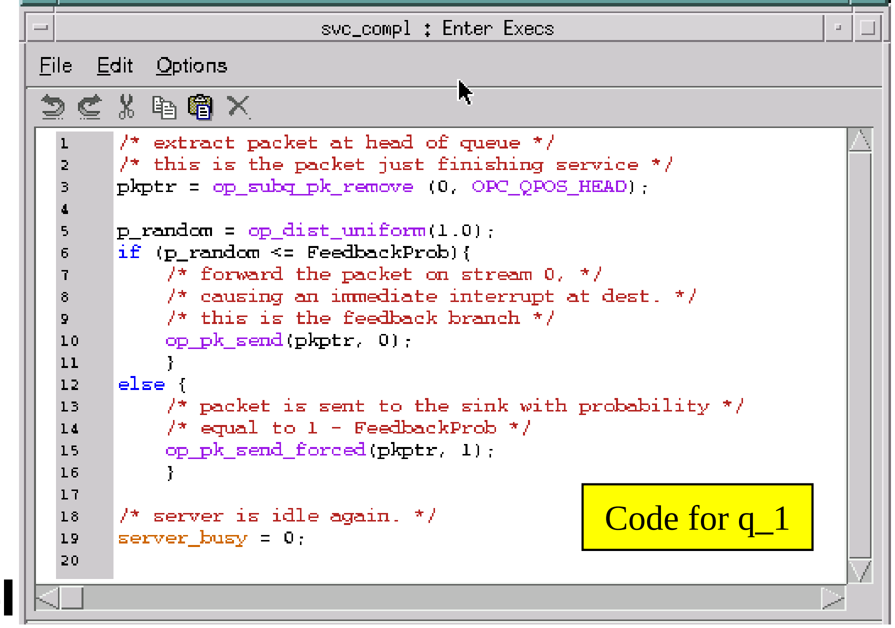


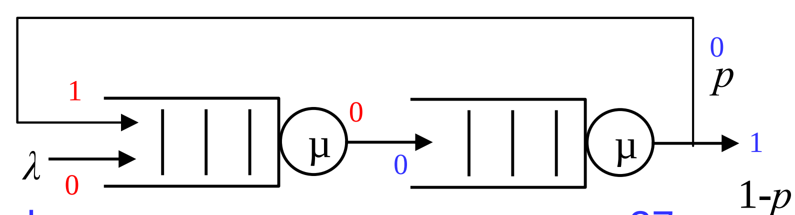
Feedback Violates the Poisson Departure process – Connectivity of Nodes – con't

- We will use the connectivity info to judge whether feedback routing code we placed in the "svc compl" state is correct for either queues or not.
- Go to the process model of node q_0 and eliminate the feedback routing. The code in the entry part of the "svc_compl" state is shown above
- Go to the process model of node q_1 and open the entry part of the "svc_compl" state. The code says that with probability "FeedbackProb" the packet is routed to output port 0 and with probability 1 - "FeedbackProb" it is routed to port 1. This matches the configuration shown below.

- Note that since the two nodes contain different process models (different code) then each has to have it own process model
  - In the process model for q_0 save the code (File/save as) and give it a name different than that used for the q_1 process model
  - In the q_0 attributes, select the new process model you've just created for q_0

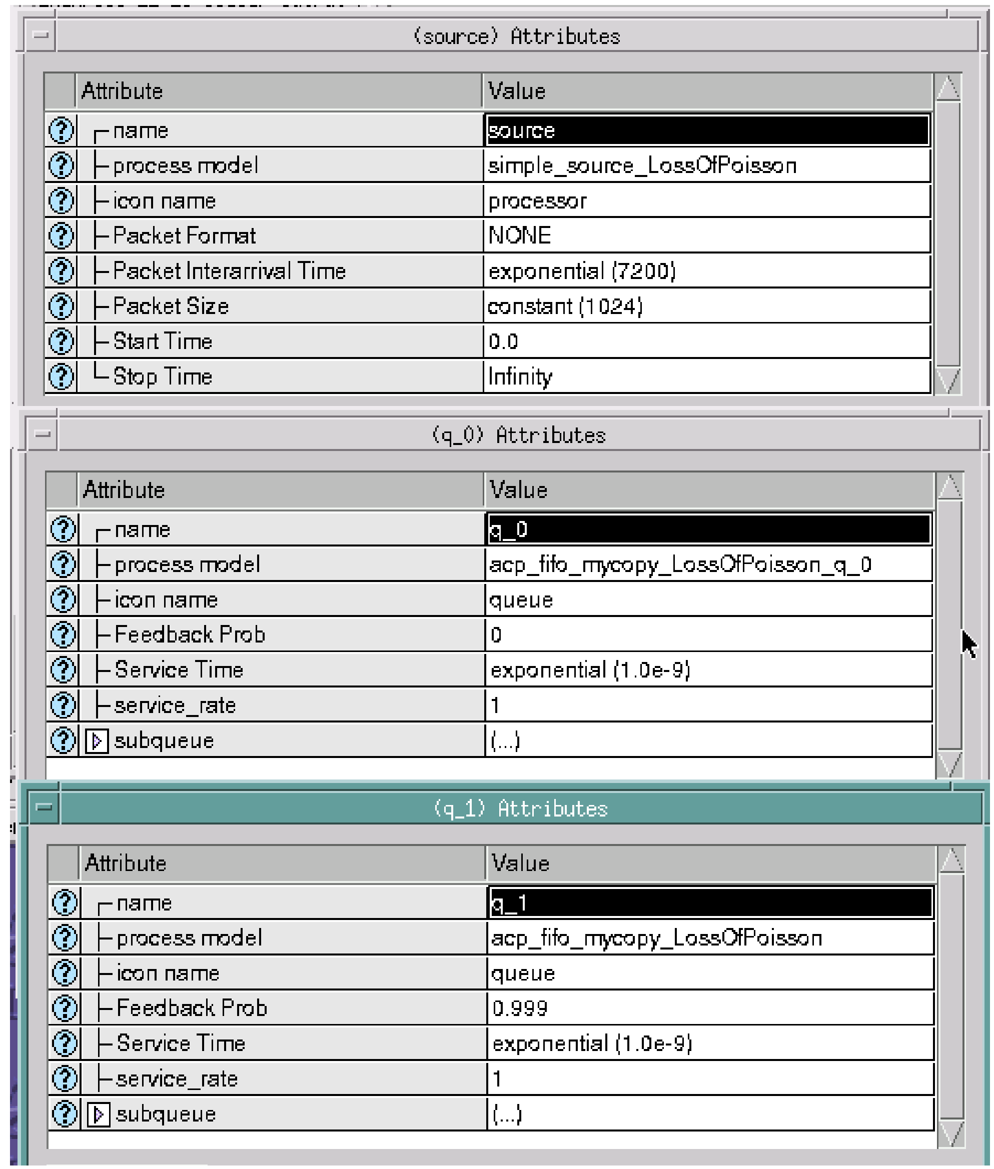






Feedback Violates the Poisson Departure process – Node Parameters

- **Right click on the each of the source, q_0, and q_1 nodes and select "Edit Attributes" to set the parameters for our simulation**
  - **Interarrival time = exponential(7200) – corresponds to 1 packet arrival per second**
  - **For q_0 the "Feedback Prob" attribute setting is irrelevant since we eliminated the use of the FeedbackProb variable in the corresponding "svc_compl" state**
  - **For q_1 set the "Feedback Prob" attribute to 0.999**
  - **For q_0 and q_1 set the "Service Time" attribute to exponential(1.0e-9)**
- **Note q_0 and q_1 have different process models**



(source) Attributes

Attribute	Value
name	source
process model	simple_source_LossOfPoisson
icon name	processor
Packet Format	NONE
Packet Interarrival Time	exponential (7200)
Packet Size	constant (1024)
Start Time	0.0
Stop Time	Infinity

(q_0) Attributes

Attribute	Value
name	q_0
process model	acp_fifo_mycopy_LossOfPoisson_q_0
icon name	queue
Feedback Prob	0
Service Time	exponential (1.0e-9)
service_rate	1
subqueue	(...)

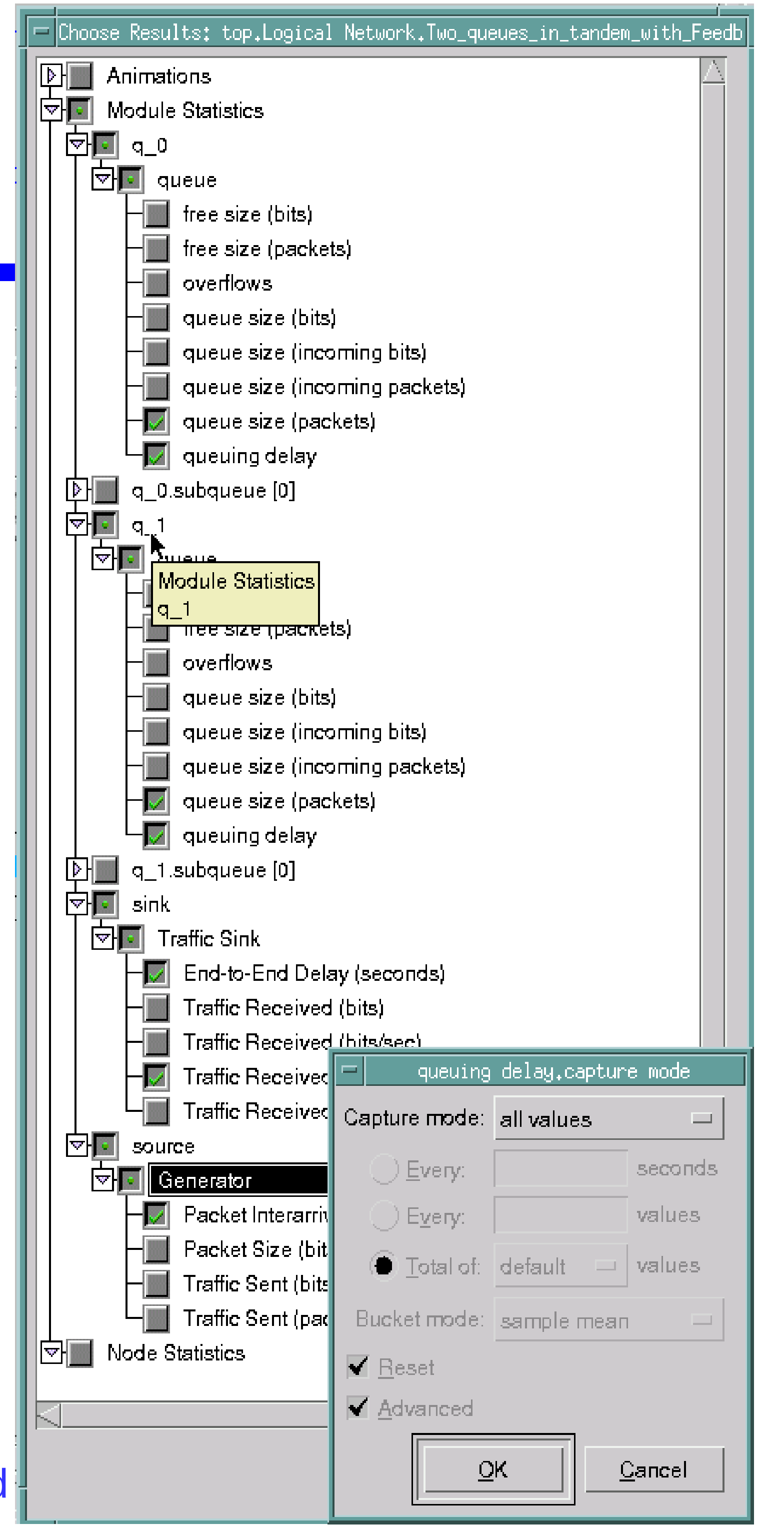
(q_1) Attributes

Attribute	Value
name	q_1
process model	acp_fifo_mycopy_LossOfPoisson
icon name	queue
Feedback Prob	0.999
Service Time	exponential (1.0e-9)
service_rate	1
subqueue	(...)

Feedback Violates the Poisson Departure process – Determining of Viewed Statistics

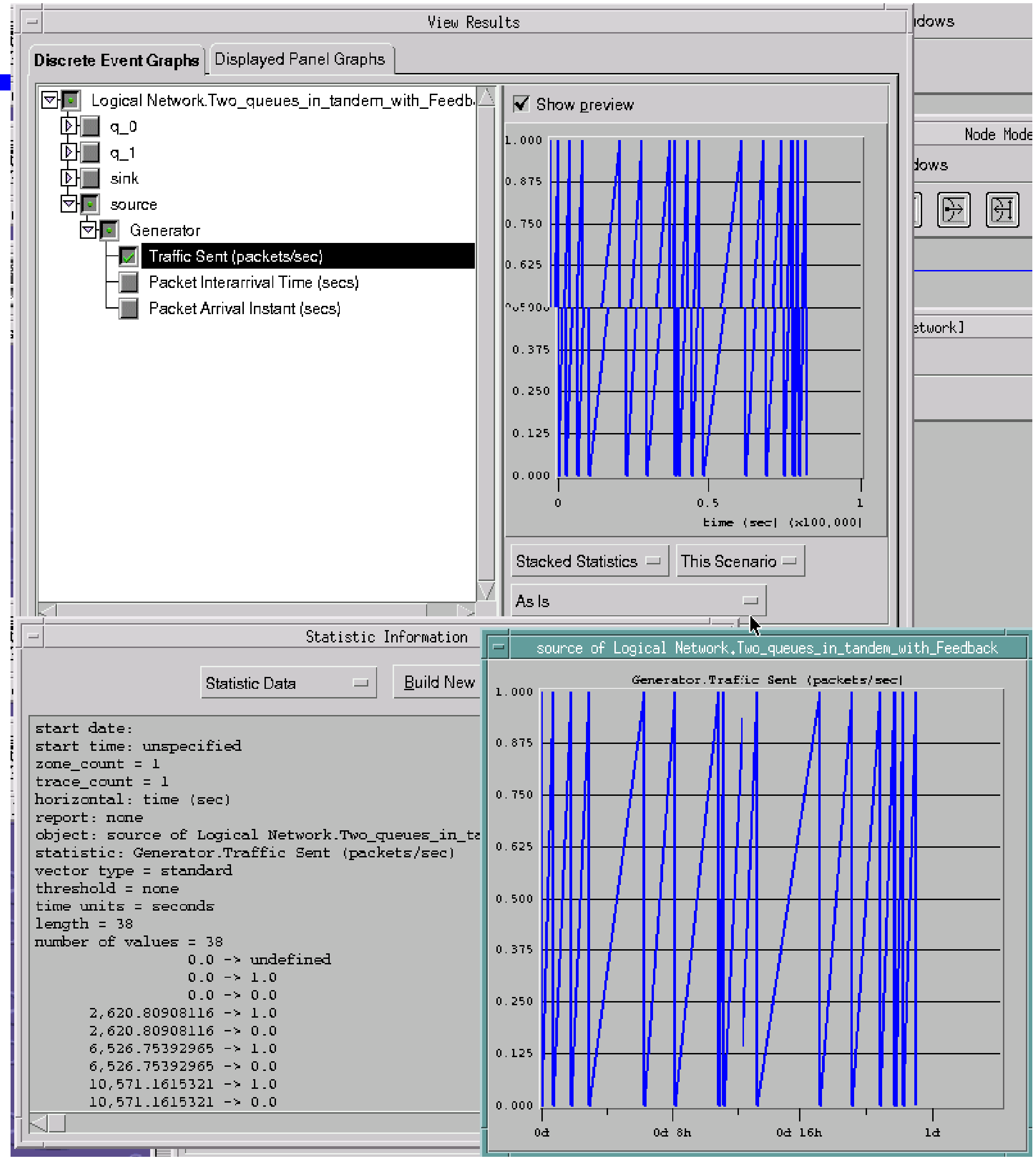
- **Right click on the project node and select "Choose Individual DES Statistics". Make the choices shown on the side figure**
- **Furthermore, since we will be interested in trying to get a timeline information similar to that on slide 45 to see arrival and departure instants of customers – we need to modify the method of collection for some of the statistics**
- **Our concern are the q_0 and q_1 delay statistics – right click on each of those statistics and select "Change Collection Mode" – in the new window – click on the "Advanced" button and change the "Capture mode" to "all values"**
  - **This means it will record all exit times for all packets without any filtering (averaging)**

- **You can do the same to "Traffic Received (packets)" statistic in the sink node to record departure instants (network departure and not queue) for customers**
- **You can do the same to "Traffic Sent (packets/sec)" statistic in the source node to record arrival instants to network for customers**



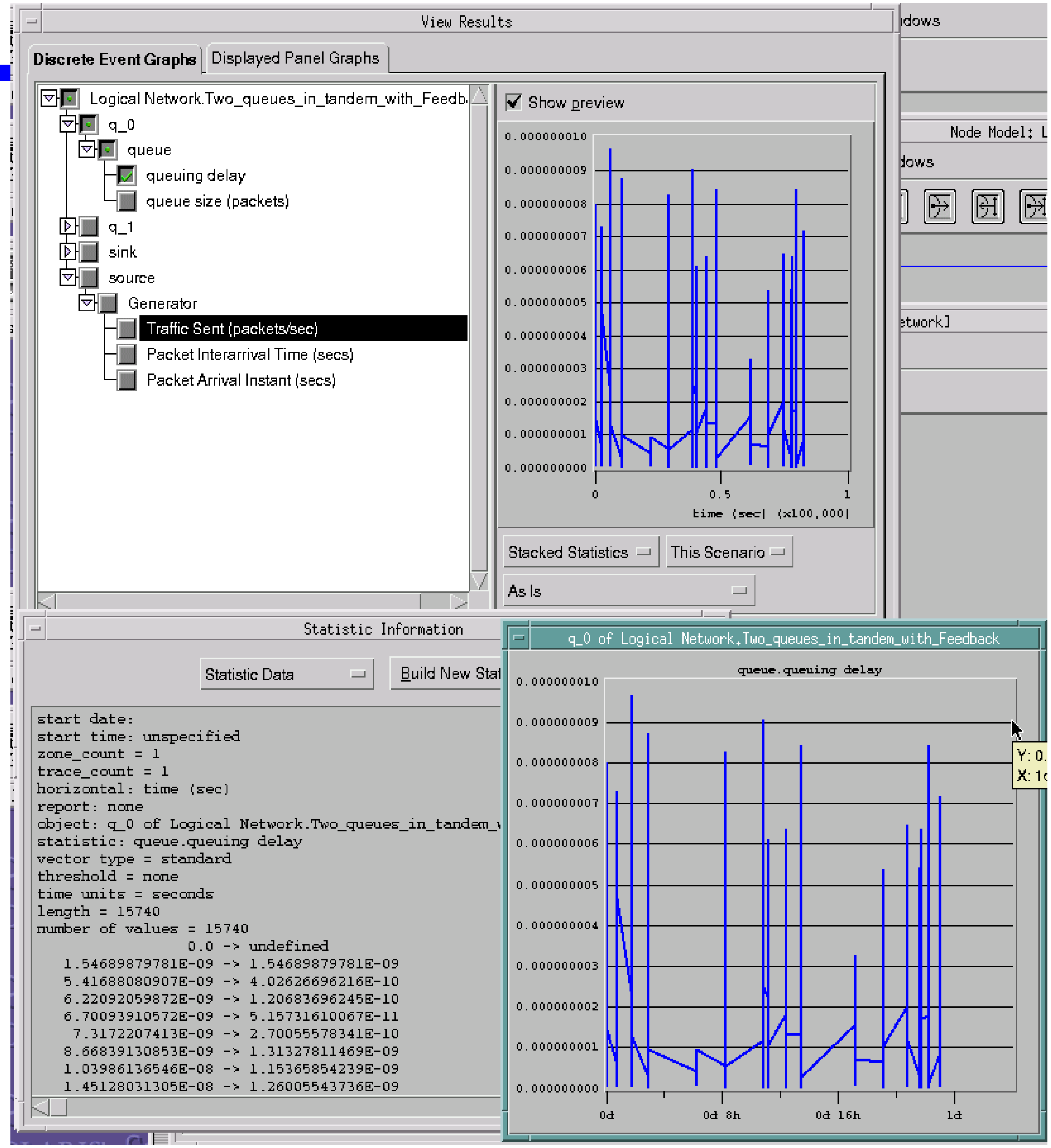
Feedback Violates the Poisson Departure process – Simulation Run

- **Save your work and press on the Configure/Run DES to run the simulation for 100,000 seconds – the simulation time is less than 20 seconds of real-time**
- **To view results right click on the project node and select view results – select to view the "Traffic Sent (packet/sec)" statistic.**
- **Click"Show" to produce a standalone graph like on the one shown on the lower right part of the figure on the side. You can also right click on the side of the graph and choose "Show Statistic Data" to display in text the statistical summary or the exact data points that are being plotted on the graphs. The figure on the side shows the statistic Data – you can see for example that the first arrival happened at t =0.0 and then at t = 2,620.8 followed by t = 6,526.7 …**



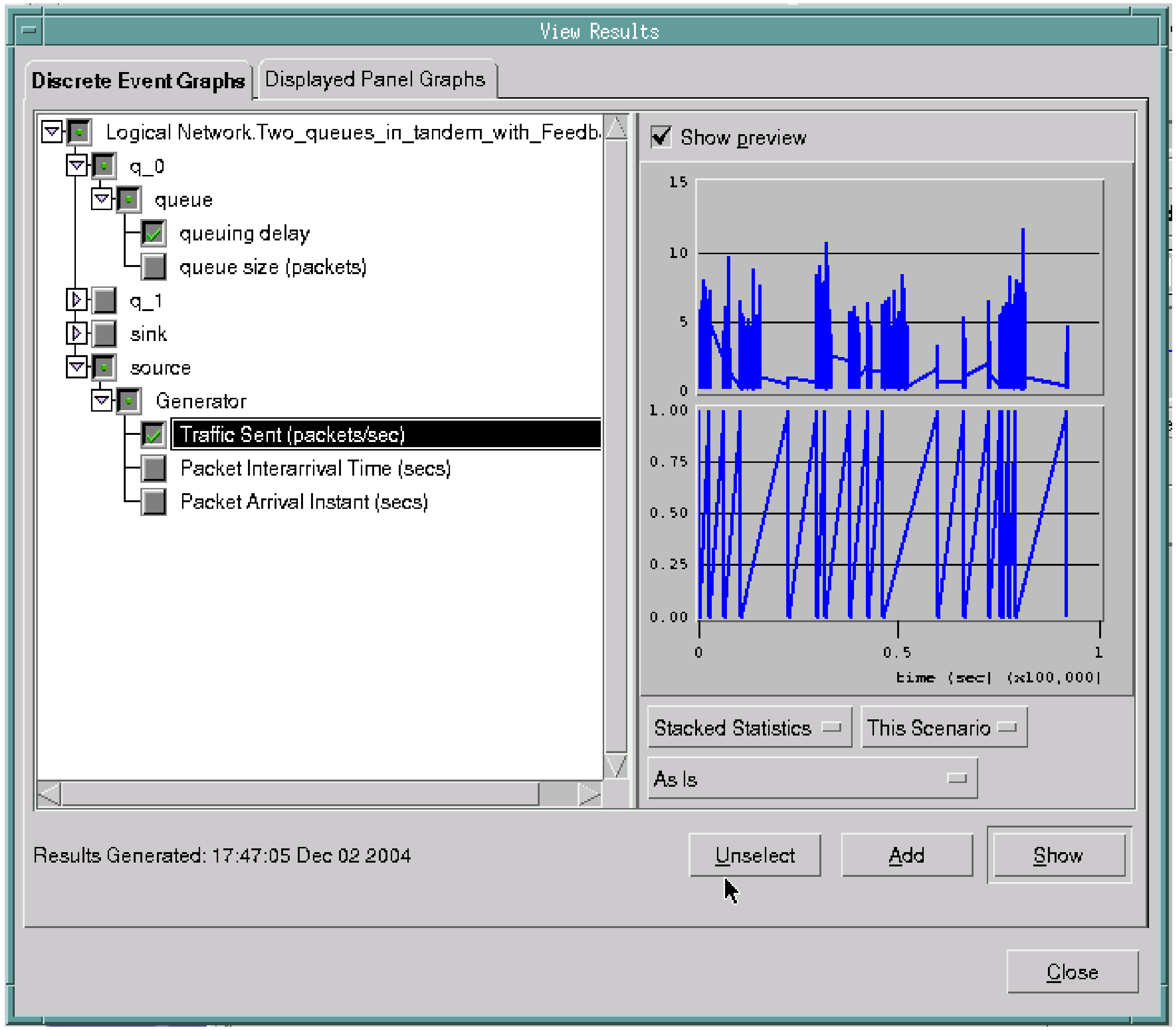
Feedback Violates the Poisson Departure process – Simulation Run (2)

- **In the same manner we can display the queuing delay for q_0 and the statistical data – that is shown in the figure on the side**
- **you can see the first packet exists q_0 at t = 1.54e-9, it circulates and comes to exit q_0 again at t = 5.41e-9, and then at t = 6.22e-9 and so on.**



Feedback Violates the Poisson Departure process – Simulation Run (3)

- To have a better visual picture – let us change the queue service rate to 1 packet / sec for both q_0 and q_1. Keep the interarrival time for the source node as exponential(7200)
- Run the simulation and display the two results for queuing delay and Traffic sent stacked as shown in the figure on the side
- You can see that for on packet sent from the source, multiple (~ 1000) recordings of the queue delay are performed.
- We can export the data we have to an excel sheet or matlab to plot a timeline similar to that on slide 45
- To export data from a graph – right click on the area left to the y-axis and select "Export All Graph Data To Spreadsheet" - this will write the data to some text file in your op_admin/tmp directory.



Feedback Violates the Poisson Departure process – Simulation Run (4)

- **Exporting the previous data and using matlab stem function to plot arrivals instants and packet delay recording instants (these are departure instants from q_0) – one can see a graph similar to the one shown on the side**
- **Although interarrival times for packets generated from the source are exponentially distributed – inter-departure times from q_0 (and same for q_1) are no longer exponentially distributed**
- **For the chosen parameters where p ≈ 1, the output of the queues is bursty and does not follow Poisson process!!**